I. **Amendments to the Claims**

Please amend the claims as follows:

1. (Currently amended) A method in a data processing system for executing instructions, the method comprising:

executing instructions in a processor in the data processing system;

detecting indicators during execution of the instructions, wherein the indicators are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with execution of the instructions or are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with accesses to memory locations;

counting events that occur within the data processing system as specified by the indicators; retrieving a count value that represents counted events; and

selecting an execution path within a set of instructions based on the count value, wherein a branch instruction that selects the execution path is contained in an object code block that contains the instructions that were executed while detecting the indicators.

2. (Original) The method of claim 1 wherein the step of selecting an execution path further comprises:

executing instructions that determine if the count value satisfies a first condition; and

branching to execute a first set of instructions in response to a determination that the count value satisfies the first condition.

3. (Original) The method of claim 2 further comprising:

branching to execute a second set of instructions in response to a determination that the count value satisfies a second condition.

4. (Original) The method of claim 1 wherein the step of selecting an execution path further comprises:

calling one of a plurality of subroutines in response to a determination that the count value satisfies a specified condition, wherein each of the plurality of subroutines is a version of a given subroutine, wherein each version has been compiled differently from other versions of the given subroutine.

5. (Currently amended) A method in a data processing system for executing instructions, the method comprising:

compiling source code statements to generate instructions for a processor in the data processing system;

generating indicators to be processed during execution of the instructions, wherein the indicators are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with execution of the instructions or are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with accesses to memory locations; and generating instructions to select an execution path within a set of instructions based on a count value that represents counted events, wherein a branch instruction that selects the execution path is contained in an object code block that contains the instructions that are executed while the indicators are processed.

6.(Original) The method of claim 5 further comprising:

processing, by a compiler, source code statements that control the selection of the execution path.

7.(Original) The method of claim 5 further comprising:

determining, by a compiler, alternative ways to compile the source code statements;

compiling the source code statements in multiple ways to generate multiple sets of instructions for the compiled source code statements; and

placing each set of instructions in a different execution path to be selected based on the count value.

8.(Original) The method of claim 7 further comprising:

identifying a compiler directive among the source code statements that specifies a type of autonomic monitoring to be implemented by the compiler; and

identifying alternative ways to compile the source code statements based on the compiler directive.

9.(Original) The method of claim 8 further comprising:

specifying an event in the compiler directive to be autonomically monitored.

10. (Original) The method of claim 5 further comprising:

determining, by a compiler, alternative ways to compile a subroutine within the source code statements;

generating a plurality of compiled versions of the subroutine, wherein each compiled version has been compiled differently from other compiled versions of the subroutine; and

placing the plurality of compiled versions of the subroutine in different execution paths to be selected based on the count value.

11. (Original) The method of claim 5 further comprising:

processing, by a compiler, compiler directives that specify the selection of the execution path.

12.(Currently amended) A computer program product on a computer readable storage medium for use in a data processing system that executes instructions, the computer program product comprising:

means for executing instructions in a processor in the data processing system;

means for detecting indicators during execution of the instructions, wherein the indicators are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with execution of the instructions or are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with accesses to memory locations;

means for counting events that occur within the data processing system as specified by the indicators;

means for retrieving a count value that represents the counted events; and

means for selecting an execution path within a set of instructions based on the count value, wherein a branch instruction that selects the execution path is contained in an object code block that contains the instructions that were executed while detecting the indicators.

13. (Original) The computer program product of claim 12 wherein the means for selecting an execution path further comprises:

means for executing instructions that determine if the count value satisfies a first condition; and

means for branching to execute a first set of instructions in response to a determination that the count value satisfies the first condition.

14. (Original) The computer program product of claim 13 further comprising:

means for branching to execute a second set of instructions in response to a determination that the count value satisfies a second condition.

15. (Original) The computer program product of claim 12 wherein the means for selecting an execution path further comprises:

means for calling one of a plurality of subroutines in response to a determination that the count value satisfies a specified condition, wherein each of the plurality of subroutines is a version of a given subroutine, wherein each version has been compiled differently from other versions of the given subroutine.

16. (Currently amended) A computer program product on a computer readable storage medium for use in a data processing system that executes instructions, the computer program product comprising:

means for compiling source code statements to generate instructions for a processor in the data processing system;

means for generating indicators to be processed during execution of the instructions, wherein the indicators are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with execution of the instructions or are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with accesses to memory locations; and

means for generating instructions to select an execution path within a set of instructions based on a count value that represents counted events, wherein a branch instruction that selects the execution path is contained in an object code block that contains the instructions that are executed while the indicators are processed.

17. (Original) The computer program product of claim 16 further comprising:

means for processing, by a compiler, source code statements that control the selection of the execution path.

18. (Original) The computer program product of claim 16 further comprising:

means for determining, by a compiler, alternative ways to compile the source code statements;

means for compiling the source code statements in multiple ways to generate multiple sets of instructions for the compiled source code statements; and

means for placing each set of instructions in a different execution path to be selected based on the count value.

19. (Original) The computer program product of claim 18 further comprising:

means for identifying a compiler directive among the source code statements that specifies a type of autonomic monitoring to be implemented by the compiler; and

means for identifying alternative ways to compile the source code statements based on the compiler directive.

20. (Original) The computer program product of claim 19 further comprising:

means for specifying an event in the compiler directive to be autonomically monitored.

21. (Original) The computer program product of claim 16 further comprising:

means for determining, by a compiler, alternative ways to compile a subroutine within the source code statements;

means for generating a plurality of compiled versions of the subroutine, wherein each compiled version has been compiled differently from other compiled versions of the subroutine; and

means for placing the plurality of compiled versions of the subroutine in different execution paths to be selected based on the count value.

22. (Original) The computer program product of claim 16 further comprising:

means for processing, by a compiler, compiler directives that specify the selection of the execution path.

23. (Currently amended) An apparatus in a data processing system that executes instructions, the apparatus comprising:

means for executing instructions in a processor in the data processing system;

means for detecting indicators during execution of the instructions, wherein the indicators are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with execution of the instructions or are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with accesses to memory locations;

means for counting events that occur within the data processing system as specified by the indicators; means for retrieving a count value that represents the counted events; and

means for selecting an execution path within a set of instructions based on the count value, wherein

<ins>a branch instruction that selects the execution path is contained in an object code block that contains the instructions that were executed while detecting the indicators</ins>.

24. (Original) The apparatus of claim 23 wherein the means for selecting an execution path further comprises:

means for executing instructions that determine if the count value satisfies a first condition; and

means for branching to execute a first set of instructions in response to a determination that the count value satisfies the first condition.

25.(Original) The apparatus of claim 24 further comprising:

means for branching to execute a second set of instructions in response to a determination that the count value satisfies a second condition.

26. (Original) The apparatus of claim 23 wherein the means for selecting an execution path further comprises:

means for calling one of a plurality of subroutines in response to a determination that the count value satisfies a specified condition, wherein each of the plurality of subroutines is a version of a given subroutine, wherein each version has been compiled differently from other versions of the given subroutine.

27. (Currently amended) An apparatus in a data processing system that executes instructions, the

apparatus comprising:

means for compiling source code statements to generate instructions for a processor in the data processing system;

means for generating indicators to be processed during execution of the instructions, wherein the indicators are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with execution of the instructions or are data values that are retrieved from memory during execution of the instructions ~~in memory~~ that specify counting of events that are associated with accesses to memory locations; and

means for generating instructions to select an execution path within a set of instructions based on a count value that represents counted events, wherein a branch instruction that selects the execution path is contained in an object code block that contains the instructions that are executed while the indicators are processed.

28. (Original) The apparatus of claim 27 further comprising:

means for processing, by a compiler, source code statements that control the selection of the execution path.

29. (Original) The apparatus of claim 27 further comprising: means for determining, by a compiler, alternative ways to compile the source code statements;

means for compiling the source code statements in multiple ways to generate multiple sets of instructions for the compiled source code statements; and

means for placing each set of instructions in a different execution path to be selected based on the count value.

30. (Original) The apparatus of claim 29 further comprising:

means for identifying a compiler directive among the source code statements that specifies a type of autonomic monitoring to be implemented by the compiler; and

means for identifying alternative ways to compile the source code statements based on the compiler directive.

31. (Original) The apparatus of claim 30 further comprising:

means for specifying an event in the compiler directive to be autonomically monitored.

32.(Original) The apparatus of claim 27 further comprising:

means for determining, by a compiler, alternative ways to compile a subroutine within the source code statements;

means for generating a plurality of compiled versions of the subroutine, wherein each compiled version has been compiled differently from other compiled versions of the subroutine; and

means for placing the plurality of compiled versions of the subroutine in different execution paths to be selected based on the count value.

33. (Original) The apparatus of claim 27 further comprising:

means for processing, by a compiler, compiler directives that specify the selection of the execution path.